Data Analytics

* What is Data Analytics:

Data Analytics is the science of analyzing raw data to find trends and answer questions.

Data analytics is the process of examining raw data to draw conclusions and make informed decisions. It involves collecting, transforming, cleaning, and analyzing data using various techniques to identify patterns, trends, and insights. The goal is to convert raw data into actionable information that can be used to improve business processes, solve problems, and predict future outcomes.

RoadMap:

1. Python: Basics of Python
2. Libraries: Numpy, Pandas, Matplotlib and Seaborn etc.
3. SQL: Data Analysis in MySQL
4. Excel: Cleaning, Transforming, Modeling, Visualization.
5. Power BI.

* Python Programming -Guido Van Rossum

What is Python: Python is a programming language. Python is both compiled and interpreted, object-oriented, high-level programming language with dynamic semantics.

Features of Python:

1. Object Oriented
2. Dynamically Typed
3. GUI programming Support
4. Extensible
5. Large Standard Library
6. Free and Open source
7. Cross platform language
8. Interpreted language
9. Expressive Language
10. Easy to Learn and use

Application of Python:

1. Data Science
2. Web Development
3. Machine Intelligence
4. Artificial Intelligence
5. Data Analytics
6. Comments in Python
7. Single Line Comments: To add single line comments, # hash is used. Python Completely ignores anything written after #.
8. Multiline line comments: To add multiline comments in python, triple quotation (“”” “”” ) are used.
9. Variables in Python:-> Variables are placeholders, which can store a value. In simple words, variable is a container that holds data inside it as a values.

Eg. a = “Hello World”

print(a) #Hello World

Rules for writing a variables:

1. Python is case-sensitive language, therefore the variables names ae case-sensitive as well.

Eg. A = “hello”

print(a) #It will throw an error as the cases used here for variable’s are different.

1. Make sure to not use spaces while creating a variable. One can use (\_) underscore to separate the names while writing a variable.
2. A variable name should never start with a number or special symbols.

* Datatypes and User-Input:

Datatypes:

1. Text-type: String (str)
2. Numeric Types: integer(int), floating point(float), complex
3. Sequence Types: list, tuple and range
4. Mapping type: Dictionaries(dict)
5. Set Type: set, frozenset
6. Boolean Type: bool
7. Binary Types: bytes, bytearray, memoryview

User-inputs: To ask for the input from the user. Default datatype is string.

Eg. name = input(“enter your name here: ”)

print(name) #enter your name here: John

age = int(input(“enter your age: ”))

print(age)

* TypeCasting and Subtypes: Conversion of one datatype to another is called as type-casting.
* There are two types of type-casting:

1. Implicit Type Conversion: where python itself converts one datatype to another.
2. Explicit Type Conversion: where the user converts one datatype to another.

* Operators and Operands:

Operators indicates what operation is to be performed while Operands indicates on what the action or the operation should be performed.

x + y = 0 in the given expression x, y, and 0 are operands and the sign(+,-,\*…etc. are operators).

Types of operators.

1. Arithmetic operators => (Addition +,Subtraction -,Multiplication \*,Floor div. //, Exponentiation \*\*, Division /, Modulus %).
2. Comparison operators => (less then / equal to ‘<’ / ‘<=’, equal to ‘==’, greater then/equal to ‘>’ / ‘>=’, not equal to ‘!=’ ).
3. Logical operators => (and, or, not.)
4. Assignment operators => (=, +=, -=, \*= )
5. Identity operators => are used to compare items to see if they are the same object with same memory address. Types 1. Is, 2. Is not
6. Membership operators => are used to check the presence of a sequence in an object type: 1. In, 2. Not in
7. Bitwise operators => These operators are used to compare the binary numbers types: 1. AND(&), 2. OR(|), 3. XOR(^), 4. Left shift(<<), 5. Right shift(>>).

* Conditional Statements: allows computer to execute a certain condition only if it is true.

Types of conditional statements.

1. if statement.
2. if-else statement.
3. if-elif-else statement.
4. nested statement.
5. short hand if statement.
6. short hand if-else statement.

Nested statement => A nested if statement is one in which an if statement is nestled inside another if statement. This is used when a variable must be processed more then once. The nested if statement in python has the following syntax.

If(condition1):

#execute if condition is true.

If(condition2):

#execute if condition 2 id true

#condition 2 ends here

#condition 1 ends here.

* Introduction to Loops:

A loop means to repeat something in the exact same way.

Types of loops are:

1. For loop => is a loop that repeats something in a given range. The range has a starting point, ending point and a step/gap in it. +1 is added to the ending point while defining a range.
2. While loop => loop executes till the given condition is true. In while loop, the increment is done inside the loop.
3. While true => It is an infinite loop, to break a while true loop, break statement is used.
4. Nested loop => A loop inside a loop is called as nested loop. Nested loops are also used to solve pattern problems.
5. For loop with Conditional Statements: The use of if-else statements increases the ability of for loop to completes the task effectively. By using if-else statements we can provide with special conditions inside for loop.

* Break and Continue Statement

Continue => is used when you want to skip a particular condition.

Break => is used when you want to destroy a loop at a certain condition and come out of the loop.

* String manipulation => Strings are the combination of number, symbols and letters, enclosed inside quotations.

Creation of string:- Strings can be created by enclosing numbers, letters or special symbols inside double quotations. It means assigning a string value toa variables.

a = (“hello world”)

print(a) // print(type(a))

String Method:- length, count, upper, lower, index, capitalize, casefold, find, format, center.

* Lists => Lists are the collection of ordered and Mutable data.

1. Lists are written inside the squared brackets.
2. The value inside a list is separated by coma(,).
3. Mutable means once created, they can be changed.
4. Multiple datatypes can be written inside a list.